Method for allocating a bandwidth between network terminals
in a communication network and network including a medium
access controller for performing such a method

5   The present invention relates to a method for allocating a bandwidth between a plurality of network terminals coupled, via a common link, to a main network station, in a communication network. It also relates to a network including a medium access controller adapted to perform the method as defined above.

10   *European patent application 0 544 975* describes a time slot management system for a time division multiple access system including a main station and a plurality of user stations via the cascade connection of a common transmission link and an individual user link. Information concerning the bandwidths respectively required for further transfer of

15   upstream information packets by the user stations are upstream transmitted to the main station by means of dedicated time slots. Registration means, located in the main station, store values indicative of said required bandwidths to have them converted in corresponding occurrences of identities, said identities being downstream transmitted

20   from the main station to the user stations. Each user station is allowed to transfer a predetermined amount of information packets to the main station upon detection of its own identity among downstream information. Such a management system does not differentiate among different service categories pertaining to different packets the network terminals

25   want to transfer to the main station and the bandwidths requested by network terminals correspond to peak rates.

         European patent application 0 957 654 describes a method to assign upstream time slots to a network terminal in a communication network including a central station and a plurality of network terminals coupled by

30   means of a cascade connection of a common link and of terminal links. Network terminals transmit upstream data packets to the central station over the common link using time slots assigned to them by means of a

downstream bitstream of network terminal grants generated by a medium access controller included in the network. Upstream data packets are classified in accordance associated service categories within network terminals and they are stored in storage queues according to their

5   respective categories. A grant related to a network terminal includes a plurality of queue grants each associated to a respective storage queue within the terminal, each queue grant enabling the said terminal to transfer a predetermined amount of upstream data packets from its corresponding storage queue. The rate of the queue grant bitstream,

10 composed of succeeding occurrences of said queue grants, is determined from at least one parameter of a parameter set associated to the corresponding storage queue and from at least one parameter of a parameter set associated to another storage queue within any of the network terminals. However, this method is not as versatile as needed

15 when applied to a network in which actual needed bandwidth changes are frequently forwarded from the network terminals to the central station and when accordingly an efficient use of the total available bandwidth is required.

A third method for allocating or sharing a bandwidth between several

20 network terminals in a network including a main station by allowance of permits, or grants, forwarded from the main station to the network terminals is to generate a stream of simple permits. Each permit corresponds to a share of the bandwidth. Upon detection by a network terminal of its own identifier included in a permit, the network terminal is

25 allowed to transfer a packet comprising a predetermined amount of information. Each network terminal cyclically forward to the main station indications concerning its required bandwidth or its load status. These indications are converted by the main station in corresponding permits. A basic example of an algorithm that can be used in this third method for

30 generating permits, is for example to define a counter for each network terminal in the main station, each counter having an associated increment value and an associated maximum. A stack or queue is included in the

main station for storing permits. Every time a permit must be forwarded to the network, the main station is programmed to firstly trigger all the counters by adding to them their associated increments and in the event a counter exceeds the maximum, stacking a permit in the stack and

5   decreasing the counter with the maximum, and secondly extracts one permit of the stack and forwards it to the network terminals.

The sharing of the available bandwidth can be modified by recalculation of the different increments, e.g. smaller increments, when no excess bandwidth is available in order to reduce the rate of permits, and

10   bigger increments for network terminals in overload, when excess bandwidth is available. This is practicable as long as the recalculation of the bandwidth share only needs to be executed when a new network terminal is installed. However, when efficient use of the total available bandwidth is required and the network terminals frequently forward actual

15   needed bandwidth changes, recalculation of the increments becomes too complex. In the following parts, the terms used will be the terms given in the explanation of the above third method.

An object of the present invention is to provide a simple method to implement a fair share of the available bandwidth including a guaranteed

20   bandwidth for each network terminal.

According to the invention this object is achieved by a method for allocating a bandwidth (BW) between a plurality of network terminals (NT_x) coupled, via a common link, to a main network station (LT) in a communication network including a medium access controller (MC) with a

25   guarantee counter (G_x) for each dependent terminal, a determined guarantee increment ($IncG_x$) and a determined guarantee maximum ($MaxG_x$) being defined for each guarantee counter and a guarantee stack (GS) for storing permits to be forwarded to the network terminals, each permit containing a network terminal identifier and corresponding to a

30   predetermined share of bandwidth, said method including for defining permits :

a first step by:

- updating the current value of each guarantee counter (G_x) by adding the corresponding guarantee increment ($IncG_x$) ;

- stacking a permit in the guarantee stack (GS) and diminishing the current value of a guarantee counter (G_x) with the guarantee maximum ($MaxG_x$), if this current value exceeds the guarantee maximum ($MaxG_x$) ;

said method being <u>characterised</u> in that the medium access controller (MC) includes also a fair counter (F_x) for each dependant terminal with a determined fair increment ($IncF_x$) and a fair maximum ($MaxF_x$) defined for each fair counter, and a fair stack (FS) for storing permits to be forwarded to the network terminals, and in that said method also includes the following two steps for defining and forwarding permits:

a second step by:

- updating the current value of each fair counter (F_x) by adding the corresponding increment ($IncF_x$) ;

- stacking a permit in the fair stack (FS) and diminishing the current value of a fair counter (F_x) with the fair maximum ($MaxF_x$), if this current value exceeds the fair maximum ($MaxG_x$) ;

a third step to forward a permit by:

- forwarding an idle permit, if the guarantee stack (GS) and the fair stack (FS) are both empty ;

- extracting a permit from the fair stack (FS) and forwarding it, if the guarantee stack (GS) is empty and the fair stack (FS) is not empty ;

extracting a permit from the guarantee stack (GS) and forwarding it, if the guarantee stack (GS) is not empty.

With such a method, the bandwidth can be shared according to the instant load status of the network terminals using computations that are simple and efficient.

An additional feature of the invention can be obtained by performing the second step only if the guarantee stack is empty, so that the fair increments have to respect a simple condition, in order to simplify their update based on the instant load status of the network terminals.

Another additional feature of the invention can be obtained by executing the second step only if the guarantee stack and the fair stack are empty, and by repeating the second step until the fair stack is not empty, so that the fair increments do not have to respect any particular
5 condition, so that their update is even more simple.

The invention also relates to, on the one hand, a communication network including a main network station coupled to a plurality of network terminals through a common link, and a medium access controller for bandwidth allocation between network terminals on the common link, said
10 medium access controller including counters and stacks for allocating and forwarding permits to network terminals, each permit containing a terminal identifier and corresponding to a predetermined share of bandwidth and, on the other hand to such a medium access controller. According to the invention, the medium access controller comprises:
15 - a guarantee counter and a fair counter for each network terminals, with a determined guarantee increment and a determined guarantee maximum for each guarantee counter and with a fair increment and a fair maximum for each fair counter;
- a guarantee stack and a fair stack for storing permits to be forwarded to
20 the network terminals,
- programmed means for defining and forwarding permits according to the method described above.

The objects and features of the invention will become more apparent and the invention itself will be best understood by referring to the
25 following description and to the accompanying drawings wherein:
- Fig. 1 is a scheme of a communication network wherein the present invention is applied;
- Fig. 2 is a block diagram of an algorithm for implementing the invention;
- Fig. 3 is a block diagram of a preferred embodiment of the algorithm;
30 - Fig. 4 is a block diagram of another preferred embodiment of the algorithm.

Figure 1 shows a communications network including a main network station LT, for instance a line terminator, which is connected to three network terminals NT_1, NT_x, NT_3, for instance three network terminators, through at least partly common transmission means. For example on figure 1, the main network station is linked to the network terminals via a tree like network comprising a common link L, connected to the main network station, and respective individual links L_1, L_x, L_3 between network terminals and common link. In such a network, permits are periodically forwarded by the main network station LT to the network terminals at a rate, which is for example one permit, every three microseconds. Each permit comprises a network terminal identifier and allows the related network terminal to transmit a predetermined amount of data. Thus, the part of the bandwidth allowed to a network terminal corresponds to the number of permits forwarded for this network terminal, divided by the total number of permits forwarded, during a predetermined period. According to the invention, the total available bandwidth **BW** is split in a guarantee bandwidth that is shared by the network terminals according to fixed weights respectively associated to the terminals, and an excess bandwidth that is fair shared according for example to the instant load statuses of the network terminals. As it will be explained, the fixed weight of a network terminal corresponds to its associated guarantee increment. In the explanations below, "x" relates to one network terminal of a network comprising "n" network terminals, and, on figure 1, to the second network terminal among three network terminals.

A medium access controller **MC** is included in the network, for instance in the main network station. It comprises a first set of counters and associated values, said set including a guarantee counter G_x for each network terminal NT_x , with an associated guarantee increment $IncG_x$ and a guarantee maximum **MaxG$_x$**, this last being possibly identical for all the guarantee counters. This first set of guarantee counters is used for stacking permits in a stack called guarantee stack **GS**, in order to share the guarantee bandwidth. The medium access controller also comprises

a second set of counters and associated values, said second set including a fair counter F_x for each network terminal, with an associated fair increment $IncF_x$ and a fair maximum $MaxF_x$, this last being possibly identical for all the fair counters. This second set of counters is used for

5   stacking permits in a stack called fair stack **FS** in order to allocate the excess bandwidth.

In the description below, triggering a counter means adding to the counter its associated increment, and if the current value of the counter is above its corresponding maximum, stacking a permit in the corresponding

10   permit stack **GS** or **FS** and diminishing the counter with its corresponding maximum. When a permit is stacked, the corresponding counter is not reset, but diminished with its corresponding maximum.

As shown in Fig. 2, forwarding permits requires three steps, the two first steps for stacking permits in corresponding stacks and the third step

15   to extract a permit of one of the stacks and forward it to the network. The medium access controller performs the first step by triggering all the guarantee counters G_x, and stacking possibly the corresponding permits in the guarantee stack **GS**. Then, a second step is started and triggers all the fair counters F_x the possible corresponding permits are stacked in the

20   fair stack **FS**. In a third step, the program analyses the status of the stacks to forward one permit in the network. If the guarantee stack and the fair stack are empty an idle permit is forwarded to the network, if the guarantee stack is empty and the fair stack is not empty a permit extracted of the fair stack and forwarded to the network, and, if the

25   guarantee stack is not empty a permit is taken out from it and forwarded to the network. According to this method, the first and second steps are performed in order to feed the permit stacks, and the third step chooses one permit by allowing the highest priority level to the guarantee stack in its choice.

30   In such a method, when the first loop is performed one time, the number of permits stacked in the corresponding stack **GS** can be zero, one, or more than one, depending on the current values of the counters,

the corresponding maximum, and on the associated increments. For example, if the set of values is:

$IncG_1$= 3, $IncGx$ =2, $IncG_3$ =1, and $MaxG_x$ = 10

When the first loop is performed ten times, it stacks six permits in the guarantee stack **GS**, which means that statistically, the global guaranteed bandwidth is 60% of the total bandwidth **BW**, and that the weights of the network terminals NT_1, NT_x, NT_3 inside the guarantee bandwidth are their respective guarantee increments $IncG_1$, $IncG_x$ and $IncG_3$. According to this example, the ratios of the physical bandwidth guaranteed to each network terminal are: NT_1: 30%, NT_x: 20%, NT_3: 10%, and 40% for the excess bandwidth. It is important to notice that according to the invention, the condition:

$$\sum_{x=1}^{n} \frac{IncG_x}{MaxG_x} \leq 1$$

must be respected in order to avoid an overflow of the guarantee stack and to perform the allocation of the guarantee bandwidth of the network terminals according to their corresponding guarantee increments.

The second loop works in the same way, the excess bandwidth is shared by the network terminals according to their respective fair increments $IncF_1$, $IncF_x$ and $IncF_3$. Moreover, the guarantee increments are fixed and defined in a configuration operation in order to allow each network terminal a fixed guaranteed bandwidth. The fair increments can be updated periodically, based on the load status of the network terminals in order to manage a fair share of the excess bandwidth. For example, the bandwidth of a loaded terminal will be increased by increasing its fair increment, and the bandwidth of a non-loaded network terminal will be decreased by decreasing its fair increment. With such a method if only one network terminal NT_x is loaded, it will get an important part of the total bandwidth **BW** allowed, said part being composed of its own part of the guaranteed bandwidth increased by the whole excess bandwidth.

In order to regulate the level in the fair stack **FS**, it is possible for example to have always the fair increments verifying the following

$$\sum_{x=1}^{n}\left(\frac{IncF_x}{MaxF_x}+\frac{IncG_x}{MaxG_x}\right)=1$$

condition :

5

Which ensures that the average rate of forwarded permits is one permit each time the algorithm is performed.

Another way to regulate the level of the fair stack is to perform the second step only if the guarantee stack is empty, as shown in the block scheme of Fig. 3. In this case, the fair increments will have to respect the condition:

$$\sum_{x=1}^{n}\frac{IncF_x}{MaxF_x}=1$$

Which is a more simple condition ensuring that statistically, each time the second step is performed, one permit is stacked in the fair stack **FS**, and which ensures also that the fair stack will not overflow.

A third approach to ensure that at the end of the second step the fair stack will remain not empty is shown in Fig. 4. This approach is to perform the second step only if the guarantee stack and the fair stack are both empty, and possibly to repeat the second step until the fair stack is no more empty. In this third approach, the fair increments do not have to respect any particular condition, in order to simplify the update of the fair increments based on the load status. With this third approach, every time the third step is started the stack status is that at least one of the stacks is not empty, and no idle grant is forwarded to the network.

It is to be noted that in the description above, the fair increments $IncF_x$ are dynamically updated according to the load status of their corresponding terminals, but the same effect can be obtained by updating

the fair maximums $MaxF_x$ and keeping the fair increments $IncF_x$ fixed. In that case, the fair maximum of a loaded terminal will be decreased to increase the corresponding permit rate, and decreased to increase the corresponding permit rate. The measurement of the load status of a terminal can be based on the queue length of a queue of this terminal, said queue length corresponding to its load status. In such a case, the queue length will be considered as if it is over a predetermined threshold value.

According to the invention, the medium access controller combines hardware and software means and more specifically counting means including the sets of guarantee and fair counters, memory means for storing the associated guarantee and fair increments and the maximal values and stacks such as for instance first in - first out stacks in order to store permits.